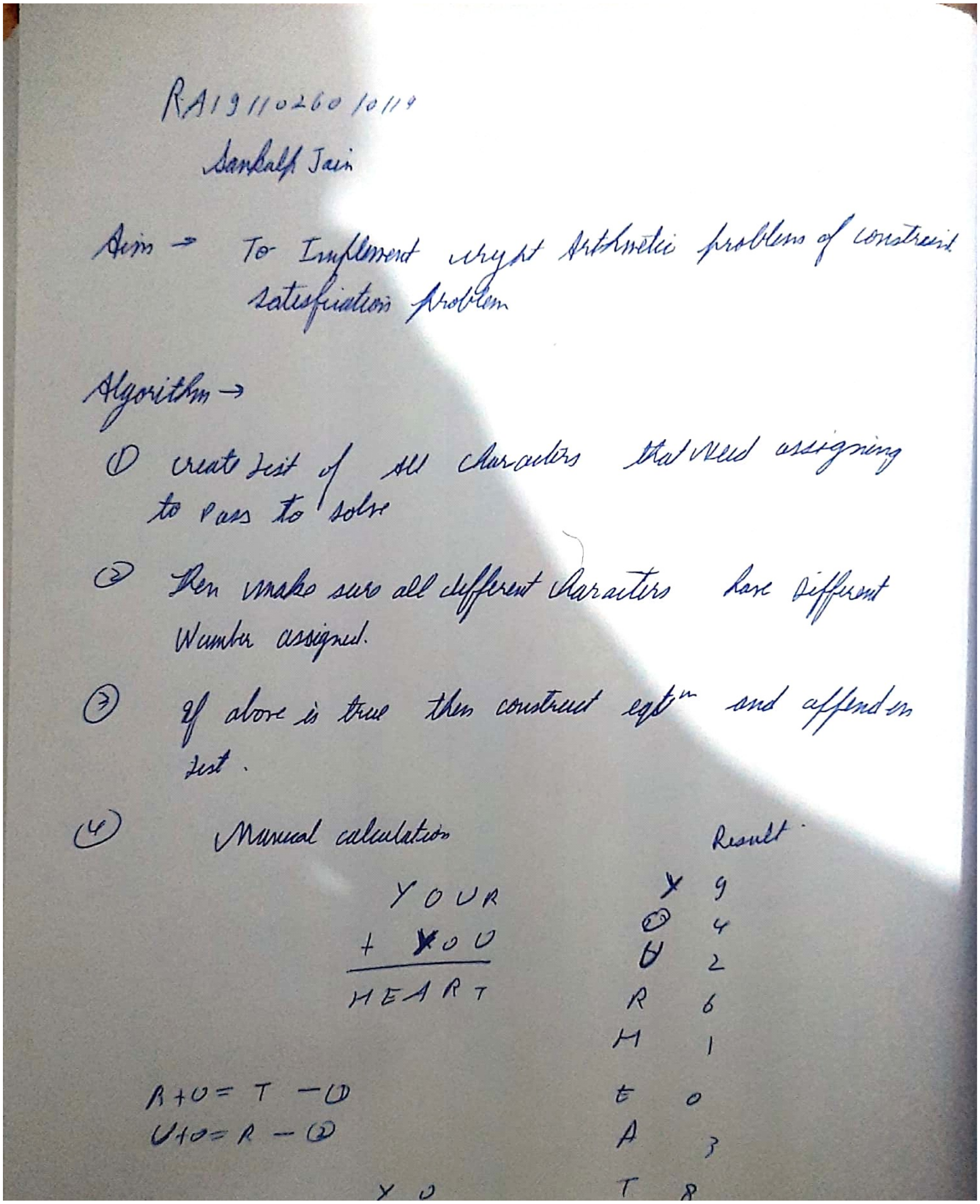
name-sankalp jain
REGno-RA1911026010119

LAB 3

Implementation of constraint satisfaction problems



RA1911026010119

Sankalp Jain

Aim → To Implement crypt Arithmetic problems of constraint satisfication problem

Algorithm →

① create list of all characters that need assigning to pass to solve

② Then make sure all different characters have different Number assigned.

③ If above is true then construct eqtn and append in list.

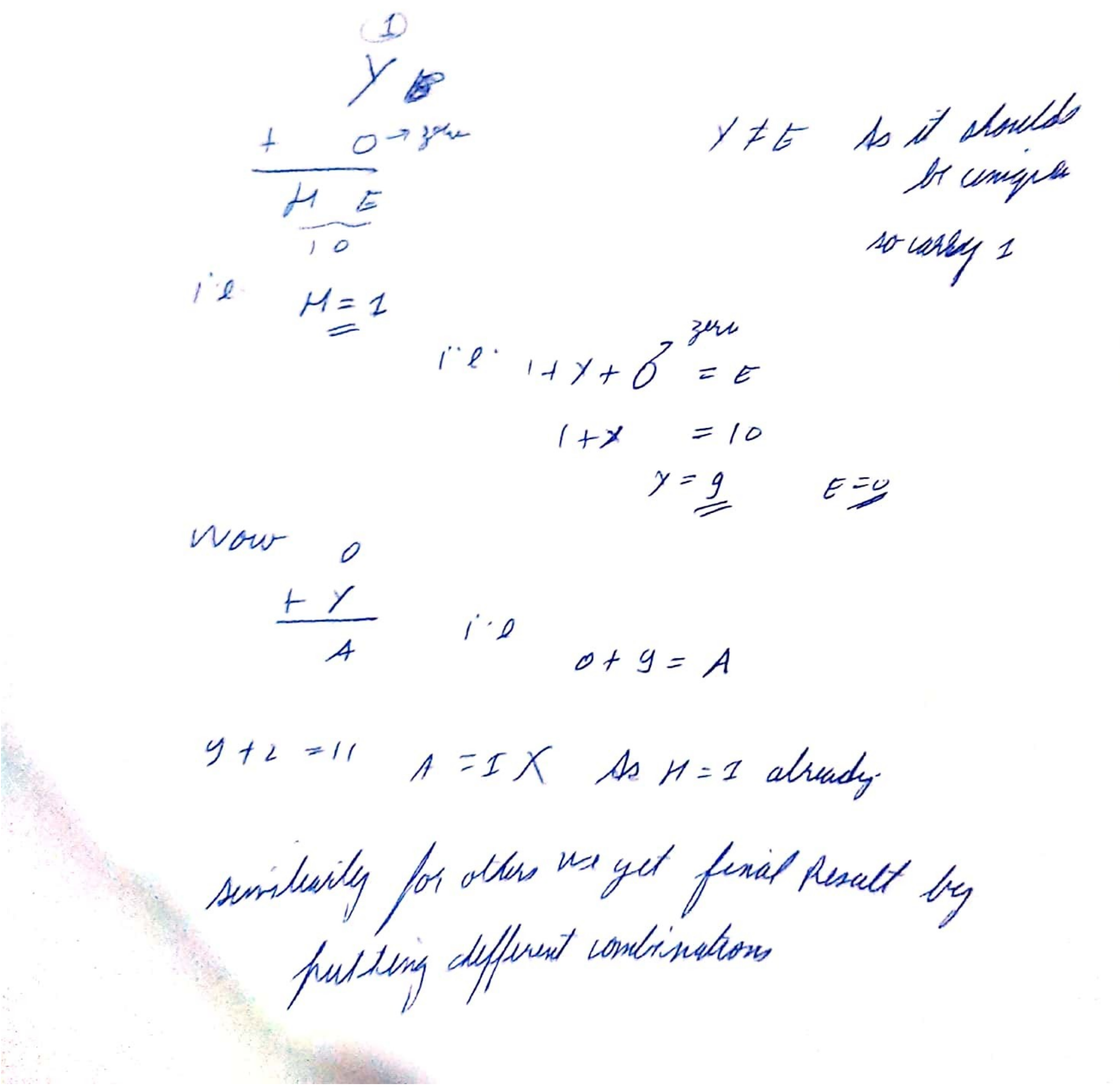
④ Manual calculation

$$\begin{array}{r} YOUR \\ +\ YOU \\ \hline HEART \end{array}$$

$R+U = T$ — ①
$U+O = R$ — ②

Y O

Result	
Y	9
O	4
U	2
R	6
H	1
E	0
A	3
T	8



①

$$\begin{array}{r} Y\ \ \\ +\ \ O \rightarrow \text{zero} \\ \hline \underbrace{H\ \ E}_{10} \end{array}$$

$Y \neq E$ As it should be unique so carry 1

i.e $\underline{\underline{H}} = 1$

i.e. $1 + Y + \overset{\text{zero}}{O} = E$

$1 + Y = 10$

$Y = \underline{\underline{9}}$ $\quad E = \underline{\underline{0}}$

Now

$$\begin{array}{r} O \\ +\ Y \\ \hline A \end{array}$$

i.e $O + 9 = A$

$9 + 2 = 11$ $\quad A = 1$ ✗ As $H = 1$ already

similarly for others we get final Result by putting different combinations

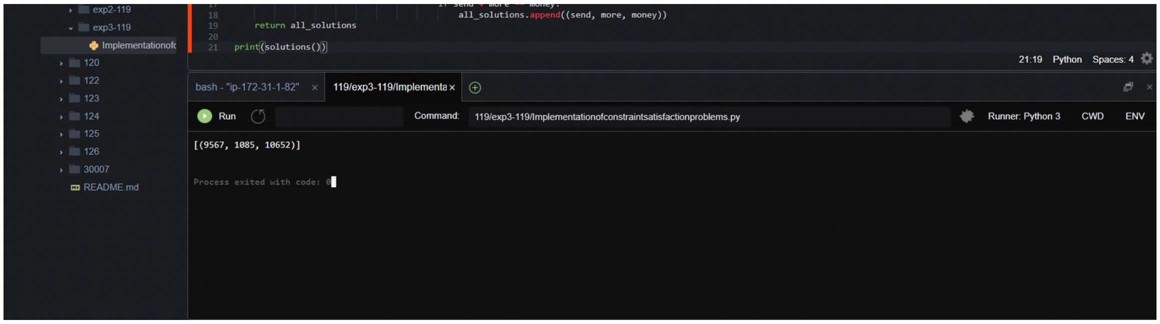
Code:

```
def solutions():

    all_solutions = list()
    for s in range(9, -1, -1):
        for e in range(9, -1, -1):
            for n in range(9, -1, -1):
                for d in range(9, -1, -1):
                    for m in range(9, 0, -1):
                        for o in range(9, -1, -1):
                            for r in range(9, -1, -1):
                                for y in range(9, -1, -1):
                                    if len(set([s, e, n, d, m, o, r, y])) == 8:
                                        send = 1000 * s + 100 * e + 10 * n + d
                                        more = 1000 * m + 100 * o + 10 * r + e
                                        money = 10000 * m + 1000 * o + 100 * n + 10 * e + y
                                        if send + more == money:
                                            all_solutions.append((send, more, money))
    return all_solutions

print(solutions())
```

OUTPUT:



```
                        all_solutions.append((send, more, money))
    return all_solutions

print(solutions())
```

Command: 119/exp3-119/Implementationofconstraintsatisfactionproblems.py

```
[(9567, 1085, 10652)]

Process exited with code: 0
```

Result:

Hence crypt algorithm problem of constraint satisfaction problem was solved and manually verified.